

# 目录

第一章：软件体系结构概论	4
1、就项目管理方面而言，软件重用项目与非重用项目有哪些不同之处。	4
2、实际参与 / 组织一个软件重用项目的开发，然后总结你是如何组织该项目的开发的？	4
3、为什么要研究软件体系结构？	4
4、根据软件体系结构的定义，你认为软件体系结构的模型应该由哪些部分组成？	5
5、在软件体系结构的研究和应用中，你认为还有哪些不足之处？	5
6、根据自己的经验，谈谈对软件危机的看法？（结合表现、原因、如何克服描述）	6
6.1、软件危机的表现？	6
6.2、软件危机的原因？	7
6.3、如何克服软件危机？	8
7、研究软件体系结构的意义是什么？	8
8、从哪些方面评估软件体系结构？	9
第二章：软件体系结构建模	10
1、选择一个规模合适的系统，为其建立“4+1...”模型。	10
2、引入了软件体系结构以后，传统软件过程发生了哪些变化？这种变化有什么好处？	12
3、软件体系结构的生命周期模型与软件生命周期模型有什么关系？	13
第三章：软件体系结构风格	13
1、层次系统结构和基于消息的层次系统结构有什么区别？	13

2、试分析和比较 B/S , 二层 C/S 和三层 C/S , 指出各自的优点和缺点。 .....	13
3、 SIS 和 DSSA 分别用在哪些场合? .....	14
4、在软件开发中, 采用异构结构有什么好处, 其负面影响有哪些? .....	15
第四章: 软件体系结构描述 .....	15
1、体系结构描述有哪些? 有哪些标准和规范? .....	15
2、体系结构描述语言与程序设计语言的区别? .....	16
3、选择一个规模适中的系统, 使用 UML 为其建模。 .....	17
第五章: 动态软件体系结构 .....	17
1、什么是动态软件体系结构? 动态软件体系结构与静态软件体系结构有什么区别? .....	17
2、基于构件的动态软件体系结构模型的层次结构是什么? .....	17
第六章: Web 服务体系结构和 SOA .....	18
1、什么是 Web 服务体系结构? 与传统的结构相比, 使用 Web 服务有哪些好处? ..	18
2、在 Web 服务中, 如何实现其松散耦合的特点? .....	18
3、试分析服务提供者、 服务请求者和服务代理三者的作用, 以及它们之间的工作流程。 .....	18
4、试解释 Web 服务栈的层次结构。 .....	19
5、Web 服务有哪些核心技术, 这些技术是如何在 Web 服务中发挥作用的。 .....	19
6、从管理的角度看, SOA 有什么优点? .....	20
7、在实际开发中, 如何实现 Web 服务和 SOA 结构? .....	20
第七章: 基于体系结构的软件开发 .....	21
1、请把基于体系结构的软件开发模型与其他软件开发模型进行比较。 .....	21

2、请把基于体系结构的软件设计方法与其他软件设计方法进行比较。	23
3、如何才能提高软件系统的可演化性。	23
第八章：软件体系结构的分析与测试	24
1、什么是软件体系结构的可靠性？为什么要研究软件体系结构的可靠性？	24
2、如何模型化系统的可靠性？	24
3、软件体系结构风险分析有哪些基本步骤？	24
4、软件体系结构测试与程序测试的主要区别是什么？	24

## 第一章：软件体系结构概论

1、就项目管理方面而言，软件重用项目与非重用项目有哪些不同之处。

答：使用软件重用技术可减少重复工作，提高软件生产率，缩短开发周期。同时，由于软构建大多经过严格的质量认证，因此有助于改善软件质量，大量使用构建，软件的灵活性和标准化程度可得到提高。

或：

软件重用项目需要考虑生产可重用软部件的机制。同时，软件重用项目可以减少开发活动中大量的重复性工作，能提高软件生产率，降低开发成本，缩短开发周期。同时由于大量使用软构件，软件的灵活性和标准化程度也可望得到提高。因此，软件重用项目在管理是比非重用项目更节约管理资源，同时管理起来更方便。

2、实际参与 / 组织一个软件重用项目的开发，然后总结你是如何组织该项目的开发的？

答：参加了一个网页管理系统的开发，该项目重复使用已有的软件产品用于开发新的软件系统，以达到提高软件系统的开发质量与效率，降低开发成本的目的。在过程中使用了代码的复用、设计结果的复用、分析结果的复用、测试信息的复用等。

3、为什么要研究软件体系结构？

答：所谓软件体系结构的分析评估，就是事先通过代价低廉的评估活动来识别软件结构中存在的潜在风险，找出软件体系结构中影响系统质量的主要因素及改进措施，并在此基础上检验软件的质量需求是否在具体设计中得到实现，并预见未来软件质量。

软件体系结构在软件开发和管理中扮演者越来越重要的角色，软件体系结构设计对软件质量有着至关重要的影响，对此最终确保系统的质量有重要的意义。软件体系结构评估，是对系统的某些值的关心的属性进行评估和判断。评估的结果可用于确认潜在的风险，并检查设计阶段系统需求的质量，在系统被实际构造之前，预测其属性质量。

或：

答：1. 软件体系结构是系统开发中不同参与者进行交流和信息传播的媒介。

2．软件体系结构代表了早期的设计决策成果。

3．软件体系结构可以作为一种可变换的模型。

4、根据软件体系结构的定义，你认为软件体系结构的模型应该由哪些部分组成？

答：构件 (component) 可以是一组代码，如程序的模块；也可以是一个独立的程序（如数据库的 SQL 服务器）；

连接件 (connector) 是关系的抽象，用以表示构件之间的相互作用。如过程调用、管道、远程过程调用等；

限制 (constrain)：用于对构件和连接件的语义说明。

5、在软件体系结构的研究和应用中，你认为还有哪些不足之处？

答：（1）缺乏同意的软件体系结构的概念，导致体系结构的研究范畴模糊。

（2）ADL 繁多，缺乏统一的 ADL 的支持。

（3）软件体系结构研究缺乏统一的理论模型支持。

（4）在体系结构描述方便，尽管出现了多种标准规范或建议标准，但仍很难操作。

（5）有关软件体系结构性质的研究尚不充分，不能明确给出一个良体系结构的属性或判定标准，没有给出良体系结构的设计指导原则，因而对于软件开发实践缺乏有力的促进作用。

（6）缺乏有效的支持环境软件体系结构理论研究与环境支持不同步，缺乏有效的体系结构分析、设计、方针和验证工具支持，导致体系结构应用上的困难。

（7）缺乏有效的体系结构复用方案。

（8）体系结构发现方法研究相对欠缺。

6、根据自己的经验，谈谈对软件危机的看法？（结合表现、原因、如何克服描述）

6.1、软件危机的表现？

软件成本日益增长

由于软件是逻辑、智力产品，软件的开发需建立庞大的逻辑体系，这是与其他产品的生产不一样的。

在软件开发过程中，用户需求变化等各种意想不到的情况层出不穷，令软件开发过程很难保证按预定的计划实现，给项目计划和论证工作带来了很大的困难。

盲目增加软件开发人员并不能成比例地提高软件开发能力。相反，随着人员数量的增加，人员的组织、协调、通信、培训和管理等方面的问题将更为严重。

开发进度难以控制

软件质量差

软件项目即使能按预定日期完成，结果却不尽人意。1965年至1970年，美国范登堡基地发射火箭多次失败，绝大部分故障是由应用程序错误造成的。

在“软件作坊”里，由于缺乏工程化思想的指导，程序员几乎总是习惯性地以自己的想法去代替用户对软件的需求，软件设计带有随意性，很多功能只是程序员的“一厢情愿”而已，这是造成软件不能令人满意的重要因素。

软件维护困难

由于在软件设计和开发过程中，没有严格遵循软件开发标准，各种随意性很大，没有完整的真实反映系统状况的记录文档，给软件维护造成了巨大的困难。

特别是在软件使用过程中，原来的开发人员可能因各种原因已经离开原来的开发组织，使得软件几乎不可维护

## 6.2、软件危机的原因 ？

### 用户需求不明确

在软件开发完成之前，用户不清楚软件的具体需求；

用户对软件需求的描述不精确，可能有遗漏、有二义性、甚至有错误；

在软件开发过程中，用户还提出修改软件功能、界面、支撑环境等方面的要求；

开发人员对用户需求的理解与用户本来愿望有差异。

### 缺乏正确的理论指导

缺乏有力的方法学和工具方面的支持。 由于软件不同于大多数其他工业产品， 其开发过程是复杂的逻辑思维过程， 其产品极大程度地依赖于开发人员高度的智力投入。 由于过分地依靠程序设计人员在软件开发过程中的技巧和创造性， 加剧软件产品的个性化， 也是发生软件危机的一个重要原因。

### 软件规模越来越大

随着软件应用范围的增广， 软件规模愈来愈大。 大型软件项目需要组织一定的人力共同完成，而多数管理人员缺乏开发大型软件系统的经验， 而多数软件开发人员又缺乏管理方面的经验。各类人员的信息交流不及时、不准确、有时还会产生误解。

软件项目开发人员不能有效地、 独立自主地处理大型软件的全部关系和各个分支， 因此容易产生疏漏和错误。

### 软件复杂度越来越高

软件不仅仅是在规模上快速地发展扩大， 而且其复杂性也急剧地增加。 软件产品的特殊性和人类智力的局限性，导致人们无力处理“复杂问题”。

所谓“复杂问题”的概念是相对的， 一旦人们采用先进的组织形式、 开发方法和工具提高了软件开发效率和能力，新的、更大的、更复杂的问题又摆在人们的面前。

### 6.3、如何克服软件危机 ？

人们面临的不光是技术问题，更重要的是管理问题。管理不善必然导致失败。

要提高软件开发效率， 提高软件产品质量， 必须采用工程化的开发方法与工业化的生产技术。

在技术上， 应该采用基于重用的软件生产技术； 在管理上， 应该采用多维的工程管理模式。

### 7、研究软件体系结构的意义是什么？

#### (1) 体系结构是风险承担者进行交流的手段

软件体系结构代表了系统公共的高层次的抽象。 这样， 系统的大部分有关人员 （即使不是全部）能把它作为建立一个互相理解的基础，形成统一认识，互相交流。

#### (2) 体系结构是早期设计决策的体现

明确了对系统实现的约束条件

决定了开发和维护组织的组织结构

制约着系统的质量属性

通过研究软件体系结构可能预测软件的质量

使推理和控制更改更简单

有助于循序渐进的原型设计

可以作为培训的基础

#### (3) 软件体系结构是可传递和可重用的模型

软件体系结构级的重用意味着体系结构的决策能在具有相似需求的多个系统中发生影响， 这比代码级重用的作用更大。

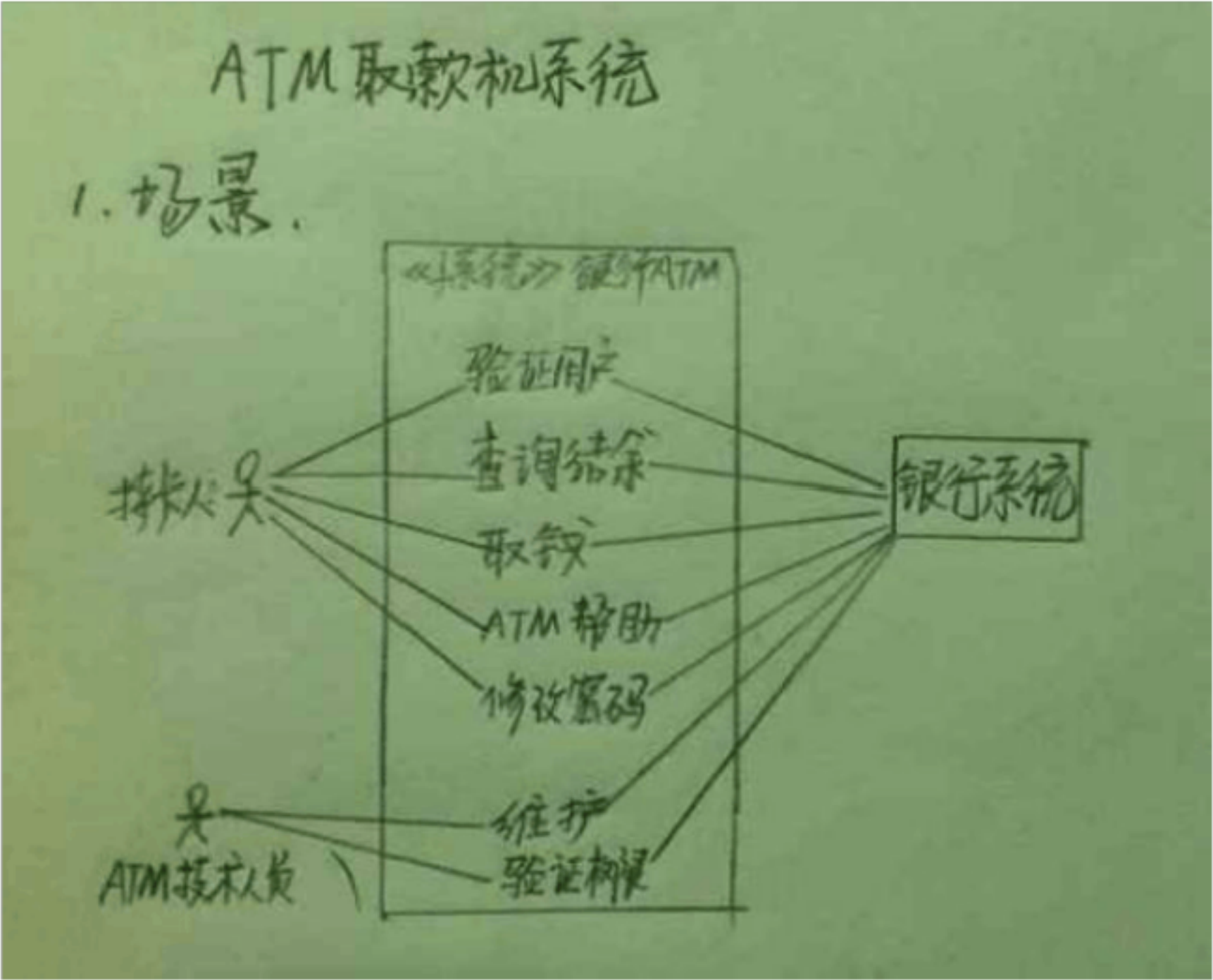


8、从哪些方面评估软件体系结构 ？

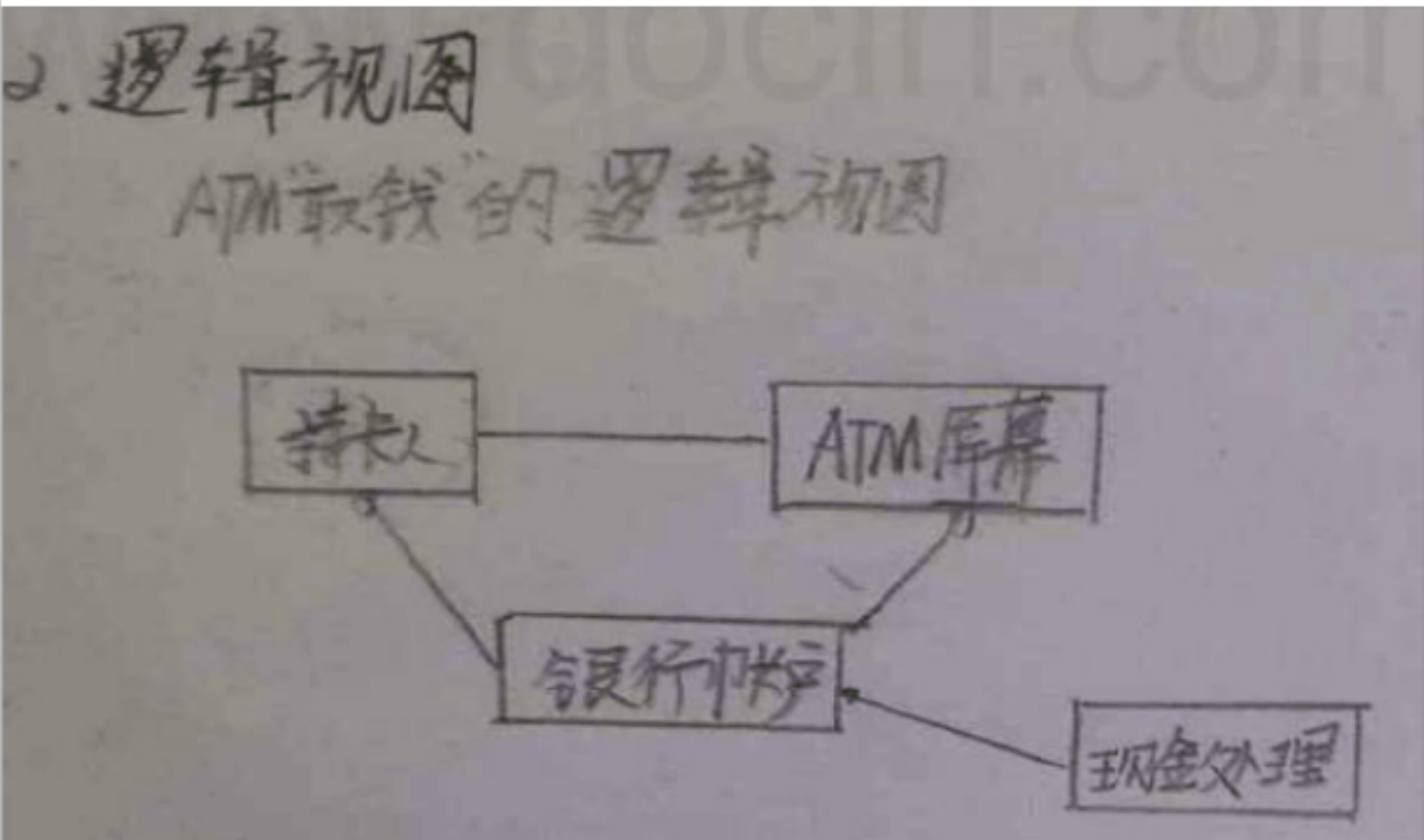
答：（ 1 ）性能是指系统的影响能力，即要经过多长时间才能对某个事件作出响应，或者在某段事件内系统所能处理的事件的个数。（ 2 ）可靠性是软件系统在应用或系统错误面前，在意外或错误使用的情况下维持软件系统的功能特性的基本能力。（ 3 ）可用性是系统能够正常运行的时间比例。 经常用两次故障间的时间长度或在出现故障时系统能够恢复正常的速度来表示。（ 4 ）安全性是指系统在向合法用户提供服务的同时能够阻止非授权用户使用的企图或拒绝服务的能力。安全性又可划分为机密性、完整性、不可否认性及可控性等特性。（ 5 ）可修改性是指能够快速的以较高的性能代价比对系统进行变更的能力。（ 6 ）功能性是系统所能完成所期望的工作的能力。（ 7 ）可变性是指体系结构经扩充或变更而成为新体系结构的能力。（ 8 ）可集成性是指系统能与其他系统协作的程度 （9）互操作性是指与其他环境或者系统本身相互作用的能力。

第二章：软件体系结构建模

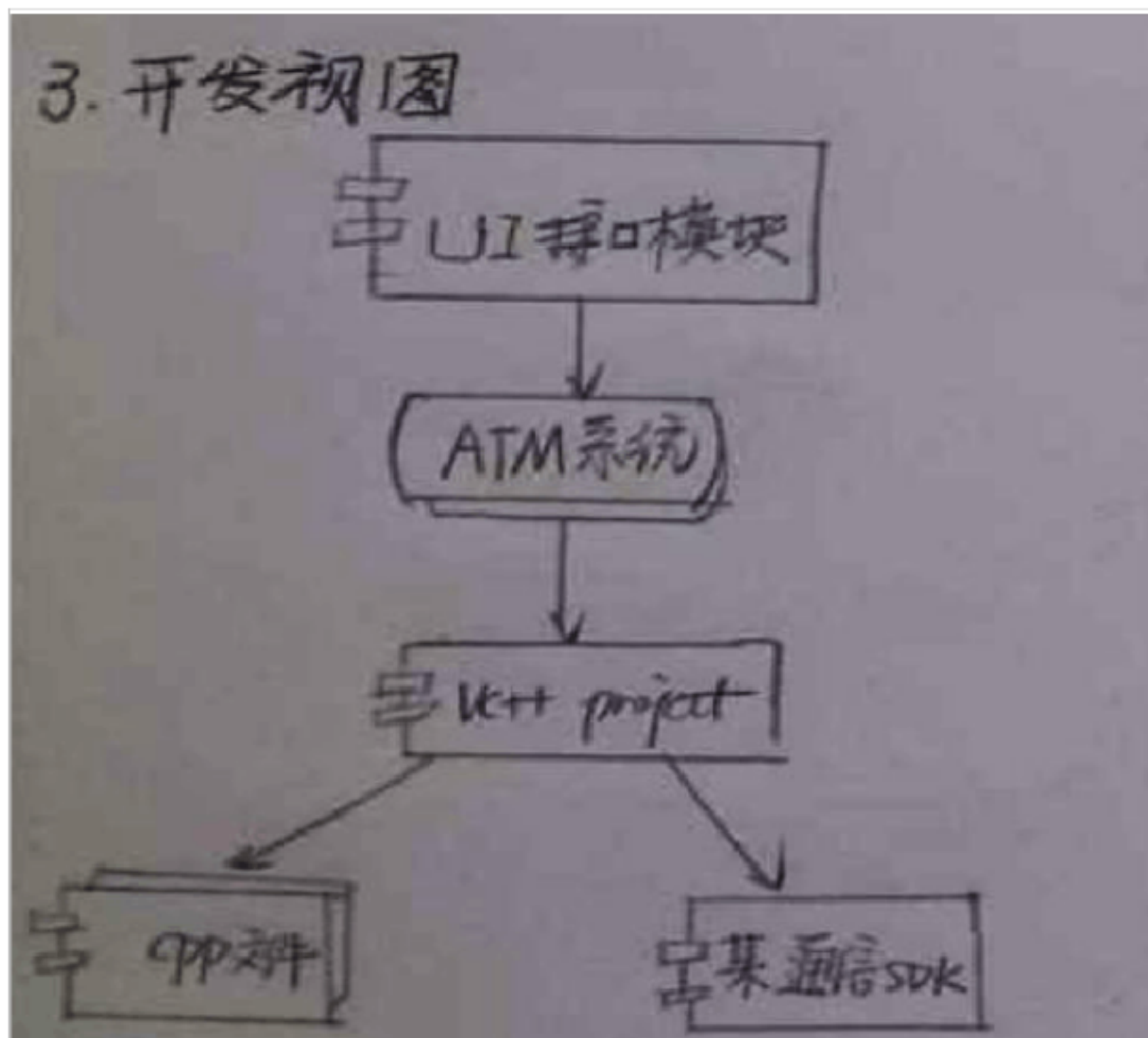
1、选择一个规模合适的系统，为其建立“4+1”模型。



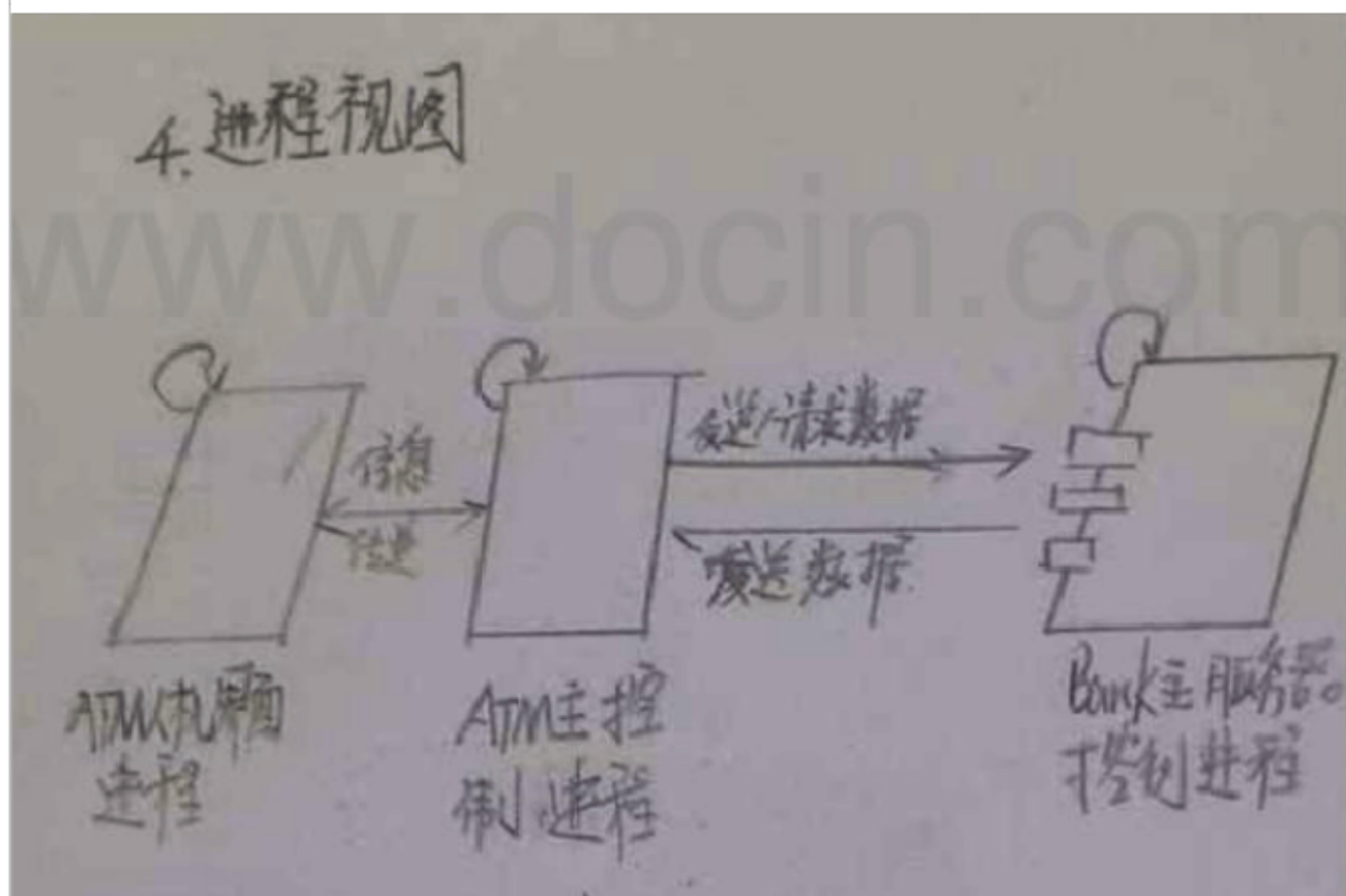
逻辑视图（ Logical View ），设计的对象模型（使用面向对象的设计方法时）。



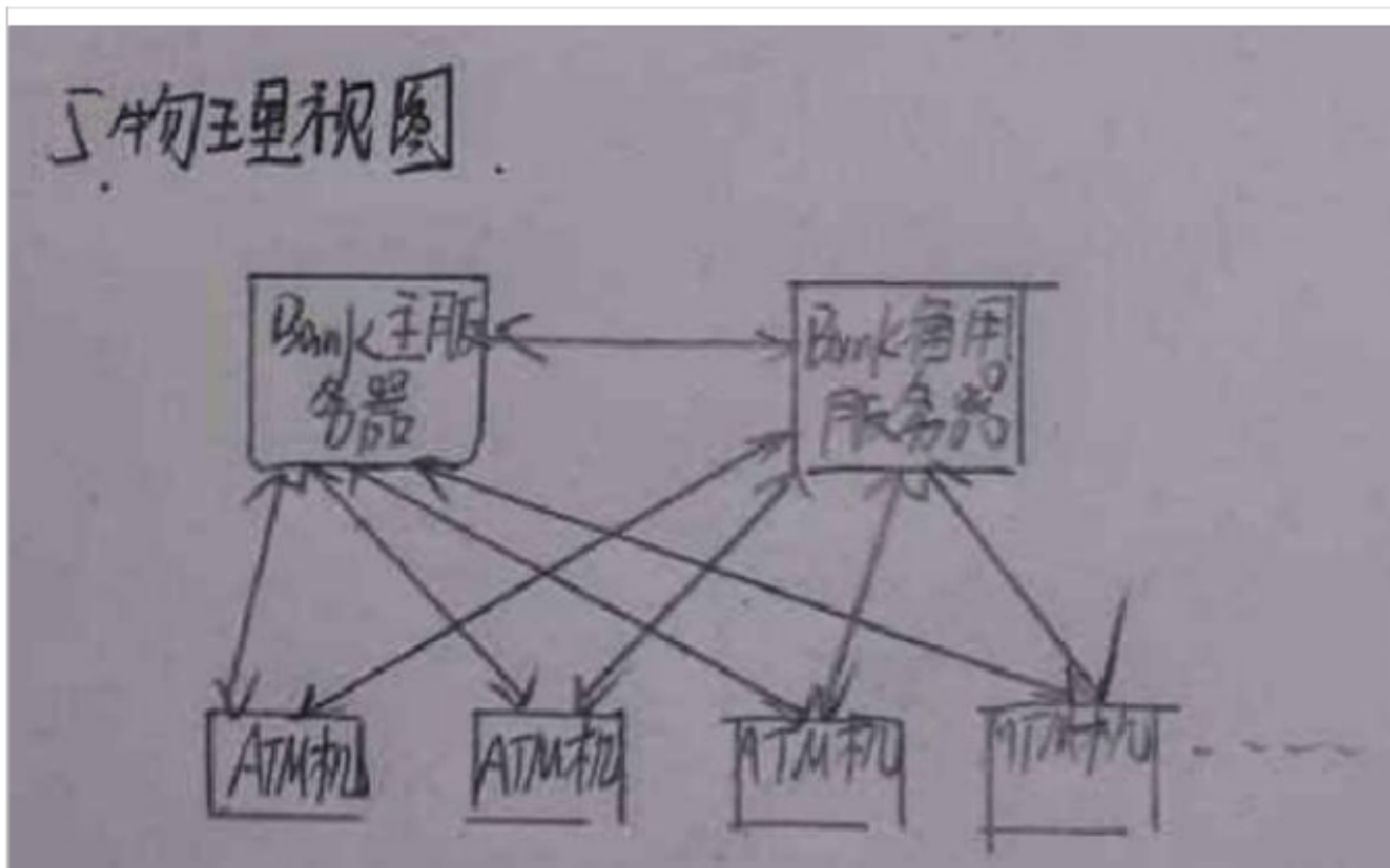
开发视图（ Development View ），描述了在开发环境中软件的静态组织结构。



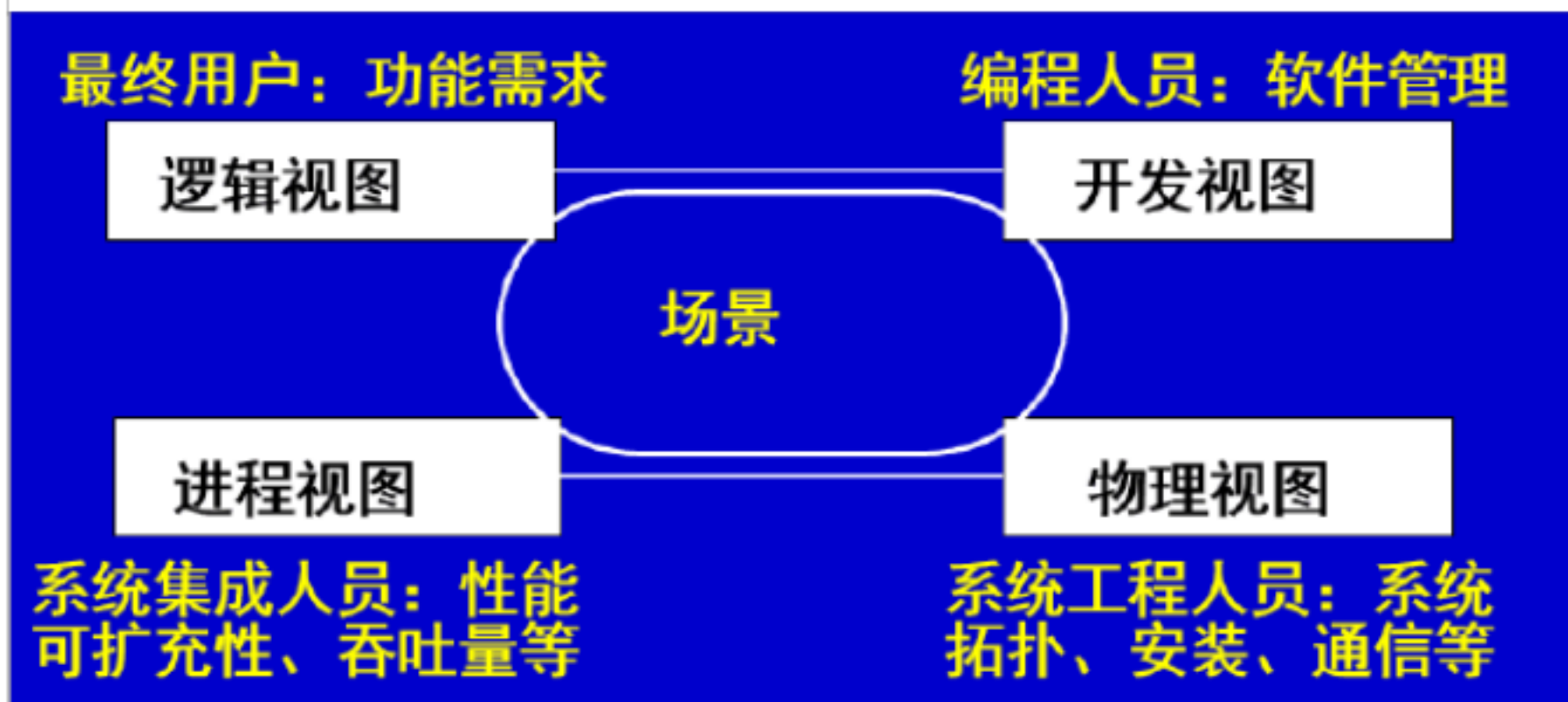
进程视图（ Process View ），捕捉设计的并发和同步特征。



物理视图（ Physical View ），描述了软件到硬件的映射，反映了分布式特性。



架构的描述，即所做的各种决定，可以围绕着这四个视图来组织，然后由一些用例（use cases）或场景（scenarios）来说明，从而形成了第五个视图。



2、引入了软件体系结构以后，传统软件过程发生了哪些变化？这种变化有什么好处？

答：软件体系结构的引入使软件设计开发更加具体和形象，它的模型更使得软件过程更加方便和多样化。其好处在于：包括程序员在内的绝大多数系统的利益相关人员都借助软件体系结构来进行彼此理解、协商、达成共识或者相互沟通的基础，软件体系结构的模型可以应用



到具有相似质量属性和功能需求的系统中，并能够促进大规模软件的系统级复用，在很多方面使得软件开发更加人性化。

### 3、软件体系结构的生命周期模型与软件生命周期模型有什么关系？

答：软件体系结构是贯穿于软件研发的整个生命周期的系统开发、运行、维护所实施的全部工作和任务的结构框架，给出了软件开发活动各阶段之间的关系，软件体系结构的生命周期模型为软件生命周期模型提供了很好的结构依据和参考，也为其构建了很好的开发方式。

## 第三章：软件体系结构风格

### 1、层次系统结构和基于消息的层次系统结构有什么区别？

答：层次结构将系统进行分级组织，其组织思想是：在层次结构中，每一层向上层提供服务，并作为客户向下层请求服务。分层系统的优点：支持基于抽象程度递增的系统设计；支持功能增强；支持重用。分层系统的缺点：并不是每个系统都可以很容易的划分为分层的模式，甚至即使是层次化的，出于性能的考虑，也不得不把一些低及或高级的功能综合起来；很难找到一个合适的、正确的层次抽象方法。

消息总线是系统的连接件、负责消息的分派、传递和过滤以及处理结果的返回。消息是构件之间通信的唯一方式。由于构件通过总线进行连接，并不要求各个构件具有相同的地址空间或局限在一台机器上，因此该风格可以很好的刻画分布式开发系统，以及 CORBA.DCOM 和 EJB 规范的系统。

### 2、试分析和比较 B/S，二层 C/S 和三层 C/S，指出各自的优点和缺点。

答：二层 C/S 体系结构将应用一分为二，服务器负责数据管理，客户机完成与用户的交互任务。优点（1）C/S 体系结构具有强大的数据操作的事务处理能力，模型思想简单，易于

人们理解和接受。（ 2 ）对软硬件的变化有极大的适应性和灵活性，易于对系统进行扩充和

缩小。（ 3 ）系统中的功能构建充分隔离，节约大量费用。缺点：（ 1 ）开发成本较高。

（ 2 ）客户端程序设计复杂（ 3 ）信息内容和形式单一（ 4 ）用户界面风格不一，使用繁杂不易推广。（ 5 ）软件移植困难（ 6 ）软件维护和升级困难（ 7 ）新技术不能轻易应用。

三层 CS 在上面的基础上进行了改造，并增加了一个服务器，其优点：（ 1 ）允许合理的划分三层结构的功能，能提高系统和软件的可维护性和可扩展性。（ 2 ）具有良好的可升级性和开放性。（ 3 ）应用的各层可以并行开发，可以选择各自最适合的开发语言。（ 4 ）为严格的安全管理奠定了坚实的基础。

B/S 风格就是上述三层应用结构的一种实现方式，其具体结构为：浏览器 /Web 服务器/ 数据库服务器。优点（ 1 ）基于 B/S 体系结构的软件，系统安装，修改和维护全在服务器端解决。（ 2 ）提供了异种机，异种网，异种应用服务的联机，联网，同意服务的最现实的开放性基础。缺点（ 1 ）缺乏对动态页面的支持能力，没有集成有效的数据库处理能力。

（ 2 ）在数据查询等响应速度上，要远远低于 C/S 体系结构。（ 3 ）数据提交一般以页面为单位，数据的动态交互性不强，不利于在线事务处理应用。

### 3、SIS 和 DSSA 分别用在哪些场合？

答：1.DSSA 只对某一个领域进行设计专家知识的提取，存储和组织，但可以同时使用多种体系结构风格；而在某个体系结构风格中进行体系结构设计专家知识的组织时，可以将提取的公共结构和设计方法扩展到多个领域。

2.DSSA 的特定领域参考体系结构通常选用一个或多个适合所研究领域的体系结构风格，并设计一个该领域专用的体系结构分析设计工具。

3 .体系结构风格的定义和该风格应用的领域是直交的，提取的设计知识比用 DSSA 提取的设计专家知识的应用范围要广。

4 . DSSA 和体系结构风格是互为互补的两项技术。

SIS：交互式网络互连服务      DSSA：特定领域的软件体系结构

4、在软件开发中，采用异构结构有什么好处，其负面影响有哪些？

答：1.结构有不同的处理能力的强项和弱点，      一个系统的体系结构应该根据实际需要进行选择，以解决实际问题。

2.软件包，框架，通信以及其他一些体系机构上的问题，目前存在者多中标准。即使再某一段时间内某一标准占据着统治地位，但变动最终是绝对的。

3.工作中，我们总会遇到一些遗留下的代码，它们仍有效用，但是却与新系统有某种程度上的不协调。然而在很多场合，将技术与经济综合进行考虑时，总是决定不重写它们。

4.在某一单位中，规定了共享共同的软件包或相互关系的一些标准，仍会存在解释或表示习惯上的不同。

负面影响：大多数应用程序只使用      10% 的代码实现系统的公开功能，      剩下 90% 的代码完成系统管理功能：输入和输出，用户界面，文本编辑，基本图表，标准对话框，通信，数据确认和旁听追踪，特定领域的基本定义等。

#### 第四章：软件体系结构描述

1、体系结构描述有哪些？有哪些标准和规范？

体系结构描述方法：

(1) 图形表达工具：由矩形框和有向线段组合而成。

(2) 模块内连接语言：将一种或几种传统程序设计语言的模块连接起来的语言。

(3) 基于软构件的系统描述语言：将软件系统描述成一种是由许多以特定形式相互作用的特殊软件实体构造组成的组织或系统。

(4) 软件体系结构描述语言：是参照传统程序设计语言的设计和开发经验，针对软件体系结构特点重新设计、开发和使用的专门的软件体系结构描述语言。

体系结构描述的标准和规范：

IEEE P1471 ：于 2000 年 9 月 21 日通过 IEEE-SA 标准委员会评审。

## 2、体系结构描述语言与程序设计语言的区别？

构造能力： ADL 能够使用较小的独立体系结构元素来建造大型软件系统；

抽象能力： ADL 使得软件体系结构中的构件和连接件描述可以只关注它们的抽象特性，而不管其具体的实现细节；

重用能力： ADL 使得组成软件系统的构件、连接件甚至是软件体系结构都成为软件系统开发和设计的可重用部件；

组合能力： ADL 使得其描述的每一系统元素都有其自己的局部结构，这种描述局部结构的特点使得 ADL 支持软件系统的动态变化组合；

异构能力： ADL 允许多个不同的体系结构描述关联存在；

分析和推理能力： ADL 允许对其描述的体系结构进行多种不同的性能和功能上的多种推理分析。

典型元素含义比较：



程序设计语言		软件体系结构	
程序构件	组成程序的基本元素及其取值或值域范围	系统构件	模块化级别的系统组成成分实体，这些实体可以被施以抽象的特性化处理，并以多种方式得到使用
操作符	连接构件的各种功能符号	连接件	对组成系统的有关抽象实体进行各种连接的连接机制
抽象规则	有关构件和操作符的命名表达规则	组合模式	系统中的构件和连接件进行连接组合的特殊方式，也就是软件体系结构的风格
限制规则	一组选择并决定具体使用何种抽象规则来作用于有关的基本构件及其操作符的规则和原理	限制规则	决定有关模式能够作为子系统进行大型软件系统构造和开发的合法子系统的有关条件
规范说明	有关句法的语义关联说明	规范说明	有关系统组织结构方面的语义关联说明

3、选择一个规模适中的系统，使用 UML 为其建模。

## 第五章：动态软件体系结构

1、什么是动态软件体系结构？动态软件体系结构与静态软件体系结构有什么区别？

答：动态软件体系结构的动态性包括：交互性动态性，结构化动态性，体系结构动态性。

由于系统需求，技术，环境，分布等因素的变化而最终造成软件体系结构的变动，称之为软件体系结构演化。 软件系统在运行时刻的体系结构变化称之为软件体系结构的动态性，动态软件体系结构的动态性包括：交互性动态性，结构化动态性，体系结构动态性。

2、基于构件的动态软件体系结构模型的层次结构是什么？

答：基于构件的动态系统结构模型支持运行系统的动态更新， 该模型分为三类， 分别是应用层，中间层和体系结构层。

（1） 应用层：处于最底层，包括构件链接，构件接口和执行

（2） 中间层：包括连接件配置，构件配置，构件描述及执行

（3） 体系结构层：位于最上层，控制和管理整个体系结构，包括体系结构配置，体

系结构描述和执行。

## 第六章：Web 服务体系结构和 SOA

1、什么是 Web 服务体系结构？与传统的结构相比，使用 Web 服务有哪些好处？

答：Web 服务作为一种新兴的 Web 应用模式，是一种崭新的分布式计算模型，是 Web 上数据和信息集成的有效机制。

Web 服务就像 Web 上的构件编程，开发人员通过调用 Web 应用编程接口，将 Web 服务集成进他们的应用程序，就像调用本地服务一样。

这主要包括一下模块内容：数据层、数据访问层、业务层、业务面、监听者；

Web 服务的不同描述：

应用的风不是；应用到应用的交互；平台无关性；

Web 服务开发生命周期：

构建、部署、运行、管理；

Web 服务的特点：

使用标准协议规范；使用协越的规范性；高度集成能力；完好的封装性；松散耦合；

与传统的结构相比，具有以下几个优点：

高度的通用性和易用性；完全的平台、语言独立性；高度的集成性；容易部署和发布；

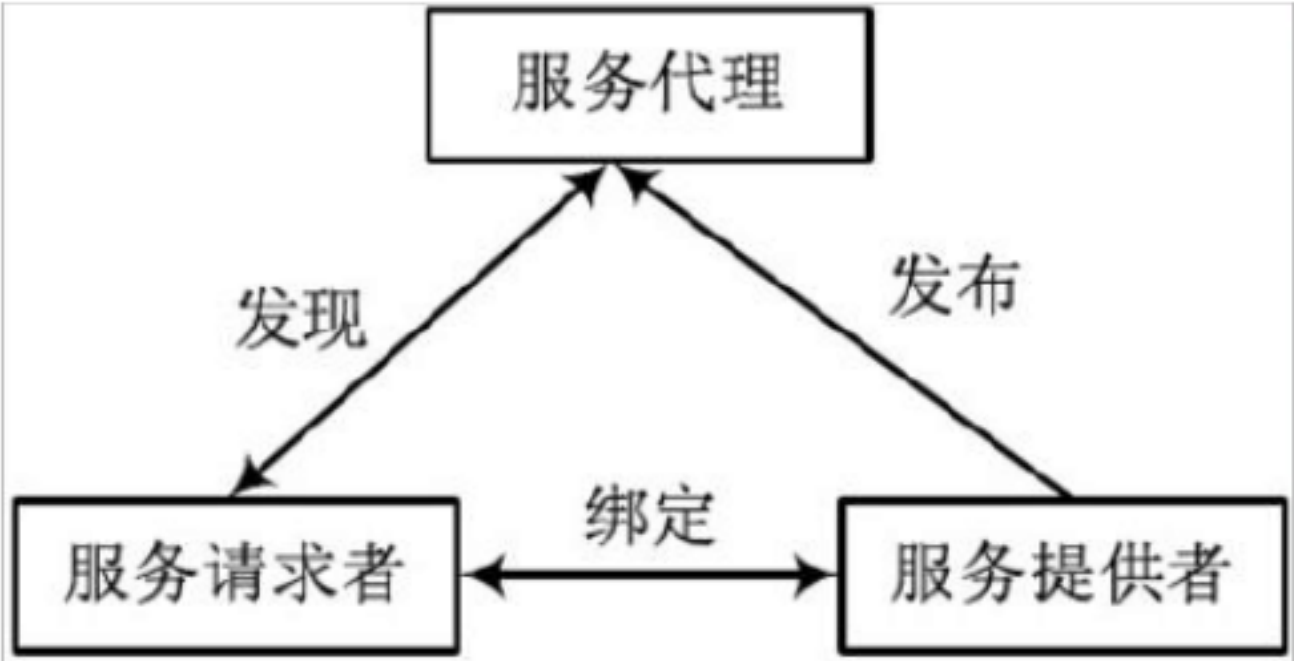
2、在 Web 服务中，如何实现其松散耦合的特点？

答：C/S 结构是松散耦合系统，它们通过消息传递机制进行通话，由客户端发出请求给服务器，服务器进行相应处理后经传递机制送回客户端。

3、试分析服务提供者、服务请求者和服务代理三者的作用，以及它们之间的工作流程。

答：服务请求者与服务提供者通过语义进行交互，服务提供者提交 web 服务描述给服务代

理者，服务代理者返回 web 服务描述给服务请求者。



4、试解释 Web 服务栈的层次结构。

答：XML( 可扩展标记语言 )、SOAP( 简单对象访问协议 )、WSDL(web 服务定义语言 )、UDDI( 统一描述发现和集成 )。

发现服务	UDDI、DISCO
描述服务	WSDL、XML Schema
消息格式层	SOAP
编码格式层	XML
传输协议层	HTTP, TCP/IP, SMTP等

5、Web 服务有哪些核心技术，这些技术是如何在 Web 服务中发挥作用的。

答：Web 服务技术核心基于可扩展标记语言 XML 的标准，包括简单对象访问协议， Web 服务描述语言和统一描述，发现和集成协议。

SOAP 定义了三部分： 定义了描述消息和如何处理消息的框架的封装， 表达应用程序定义的数据类型实例的编码规则以及描述远程调用和应答的协议和 SOAP 编订。

WSDL 为服务者提供以 XML 格式描述的 WEB 服务请求的标准格式，经网络服务描述为能

够进行消息交换的通信端点集合，以表达一个 Web 服务能做什么，他的位置在哪里，以及如何调用等信息。

UDDI 规范描述了 Web 的概念，同时也定义了一种编程接口。通过 UDDI 提供的标准接口，企业可以发布自己的 Web 服务供其他企业调用和查询，业可以查询特地服务的描述信息，并动态的绑定到该服务上，通过 UDDI，Web 服务可以真正实现信息的“一次注册到处访问”。

6、从管理的角度看，SOA 有什么优点？

答：（1）更易于维护；服务提供者和服务和服务使用者的松散耦合关系及对开放标准的采用确保了该特性的实现。

（2）更高的可用性；该特性在服务提供者和服务使用者的松散耦合关系上得以体现。使用者无需了解提供者的实现细节。

（3）更好的伸缩性；依靠服务设计、开发和部署所采用的架构模型实现伸缩性。服务提供者可以彼此独立调整，以满足服务需求。

7、在实际开发中，如何实现 Web 服务和 SOA 结构？

答：声明技术：J2EE 编程模型就是使用声明技术提供应用程序逻辑和中间件配置分离的一个例子。

抽象：在某些情况下，SOA 基础结构中可以提供 API，以用于特定的用途。例如，SOA 基础结构可以提供错误报告和审核机制。在设计此类 API 时应非常小心，要注意其易用性。

我们应优先考虑声明技术，而不是对这些机制进行编程配置。同样，在标准 API 可用时，我们应通过这些标准 API 公开 SOA 基础结构功能，而不是采用自己开发编写的方式。

代码生成：在无法避免代码复杂性的地方，可以使用代码生成技术。例如，Web 服务描述语言（Web Services Definition Language，WSDL）就可以为开发人员隐藏 SOAP、

HTTP 和 JMS 的复杂细节。这是通过组合用 WSDL 表示的可由计算机处理的接口定义和可从 WSDL 生成相关调用代码的语言特定实现的工具来实现的。

工具：在不可避免 SOA 基础结构的细节进入开发人员代码的情况下，我们可以通过使用合适的工具扩展开发环境来减少开发人员工作的复杂性。

IBM Rational? Software Development Platform 产品所提供的基于 Eclipse 的环境可使用自定义插件、代码片段和用户指南轻松地进行扩展。

模型驱动的开发：模型驱动的开发技术可以被视为前面两种方法的特定复杂组合，同时利用了工具和代码生成功能来简化开发体验。开发人员生成统一建模语言（Unified Modeling Language，UML）模型，此类模型可转换为相应的代码，其中包含利用 SOA 基础结构所必需的代码。

总之，在定义面向服务的体系结构及其基础结构时，我们必须特别注意开发人员的需求。当为开发人员提供指南，以告知他们应如何开发或使用服务时，我们应该寻找可促进这些指导方针遵循的机制。SOA 内的控制对其成功甚为关键。

## 第七章：基于体系结构的软件开发

### 1、请把基于体系结构的软件开发模型与其他软件开发模型进行比较。

答：软件开发模型有演化模型、螺旋模型、喷泉模型、智能模型等。传统软件开发模型存在开发效率不高，不能很好地支持软件重用等缺点。在多个大中型软件项目的实践基础上，提出了基于体系结构的软件开发模型（ABSD）。ABSD 模型把整个基于体系结构的软件过程划分为体系结构需求、设计、文档化、复审、实现、演化等 6 个子过程，讨论了各个子过程所要完成的工作，给出了 ABSD 模型在劳动和社会保险领域的一个应用实例。实践表明，采用 ABSD 模型进行软件项目开发，具有结构清晰、易于理解、可移植性强、重用粒度大等优点。

答：一、基于体系结构为基础的基于构件组装的软件开发模型，如基于构件的开发模型和基于体系结构的开发模型等。基于体系结构的开发模型是以软件体系结构为核心，以基于构件的开发方法为基础。然后采用迭代增量方式进行分析和设计，将功能设计空间映射到结构设计空间，再由结构设计空间映射到系统设计空间的过程。

特点：是利用需求分析结果设计出软件的总体结构，通过基于构件的组装方法来构造软件系统。

优点：软件复用，提高了开发的效率。软件的结构框架更清晰，有利于系统的设计、开发和维护。

缺点：缺乏通用的组装结构标准，因而引入了较大的风险。可重用性和软件高效性不易协调。客户的满意度低，过分依赖于影响产品质量。

二、以软件需求完全确定为前提软件开发模型，如瀑布模型等。

特点：软件需求在开发阶段已经被完全确定，将生命周期的各项活动依顺序固定，强调开发的阶段性；

优点：开发流程简单。

缺点：是开发后期要改正早期存在的问题需要付出很高的代价，用户需要等待较长时间才能够看到软件产品，增加了风险系数。

三、在开始阶段只能提供基本需求的渐进式开发模型，如螺旋模型和原型实现软件开发模型等。

特点：软件开发开始阶段只有基本的需求，软件开发过程的各个活动是迭代的。通过迭代过程实现软件的逐步演化，最终得到软件产品。在此引入了风险管理，增加项目成功几率

优点：强调了其他模型均忽略了的风险分析。

缺点：由于需求的不完全性，从而为软件的总体设计带来了困难并要求对风险技能管理水平

的高要求。

2、请把基于体系结构的软件设计方法与其他软件设计方法进行比较。

基于体系结构的开发方法出现了基于构件、基于框架、基于模板的开发技术，项目利用基于体系结构开发的思想，从需求到高层框架设计、架构设计、平台实现，充分体现了基于体系结构开发中从高层设计入手，复用体系结构框架、设计和实现可复用的构件的原则。

此外，与其他软件设计方法的区别还有：

1、单个用户界面和业务逻辑的改动不会影响到系统整体

2、开发人员的组织用户界面和业务逻辑的定制可由领域人员来完成；专业

计算机人员制作不同粒度的构件，完成模型的开发。

3、不同业务对象有相似的业务逻辑，可直接复用其构件。4、在系统维护过程更新某一模块，不会影响系统的其它部分。

3、如何才能提高软件系统的可演化性。

答：构造性和演化性是软件的两个基本特性。软件进行渐变并达到所希望的形态就是软件演化，软件演化是由一系列复杂的变化活动组成。对软件变化的控制是软件开发者历来追求的目标。引起软件变化的原因是多方面的，如基本设施的改变，功能需求的增加，高性能算法的发现，技术环境因素的变化等。所以对软件变化甚至演化进行理解和控制显得比较复杂和困难。

(1) 必须频繁地变化以适应要求。

(2) 软件的复杂度不断地增长。

(3) 通过自我调节以符合产品需求和过程特性。

(4) 在软件的生命周期中保持一定的组织稳定性。

(5) 不同的版本之间保持一定的连贯性。(6) 功能持续地增加。

(7) 在没有严格的维护和适应性修改的情况下会出现质量衰退。 (8) 是一个反馈系统。

## 第八章：软件体系结构的分析与测试

1、什么是软件体系结构的可靠性？为什么要研究软件体系结构的可靠性？

答：通过系统的详细说明书，确定系统所采用的体系结构风格。

把每一种体系结构风格转换成状态视图，并计算状态视图中每一个状态的可靠性及其相应的迁移概率。

通过整个系统的体系结构视图，把所有的状态视图集成为一个整体状态视图。

通过整体状态视图构造系统的迁移矩阵，并计算系统的可靠性。

2、如何模型化系统的可靠性？

答：软件体系结构风险分析有哪些基本步骤

采用体系结构描述语言 ADL 对体系结构进行建模 通过模拟方法执行复杂性分析

通过 FMEA 和模拟运行执行严重性分析 为构件和连接件开发其启发式风险因子 建立用于风险评估的 CDG

3、软件体系结构风险分析有哪些基本步骤？

答：软件体系结构风险分析有哪些基本步骤

采用体系结构描述语言 ADL 对体系结构进行建模 通过模拟方法执行复杂性分析

通过 FMEA 和模拟运行执行严重性分析 为构件和连接件开发其启发式风险因子 建立用于风险评估的 CDG

4、软件体系结构测试与程序测试的主要区别是什么？

答：软件体系结构测试要研究的对象是软件体系结构设计， 并以此为基础产生高层次的测试

用例集，以指导代码层的测试活动。 他的目的是找出体系结构设计的错误和缺陷， 产生指导



代码测试的测试计划和测试用例， 软件体系结构测试与传统的软件测试既有联系又有区别， 而软件体系结构的测试包含两个方面， 一个是软件体系结构的结构分析， 一个侍女软件体系结构的测试。

程序测试是指对一个完成了全部或部分功能、 模块的计算机程序在正式使用前的检测， 以确保该程序能按预定的方式正确地运行。

目前， 软件的正确性尚未得到根本的解决， 软件测试仍是发现软件错误和缺陷的主要手段。

为了发现系统中的错误， 应竭力设计能暴露错误的测试用例。 测试用例是有测试数据和预期结果构成的。 一个好的测试用例是极有可能发现至今为止尚未发现的错误的测试用例。

高效的测试是指用少量的测试用例， 发现被测软件尽可能多的错误。 软件测试所追求的是以尽可能少的时间和人力发现软件产品尽可能多的错误。